

## **Performance Tuning at CM Loader Program While Replicating EQP List for IBM SiView**

### **Background of the Invention**

#### **1. Field of Invention**

This invention relates generally to the relational database management system (RDBMS) data replication, and more particularly, to data replication between relational databases used in a semiconductor manufacturing channel monitoring environment where performance has a high impact on productivity.

#### **2. Description of Related Art**

There are many kinds of methods to replicate data from RDBMS to RDBMS, including triggers, stored procedure, and DB logs. No matter which method is used, the outcome is the same whether it be a simple IBM DB2 to DB2 homogeneous replication or a more complex replication between heterogeneous database types. For a more involved requirement such as a data pre-summary that is used to reduce the amount of data that must be accessed, a loader program is useful as its programming language affords more power for computing and doing data conversion.

In the semiconductor manufacturing Fab environment a loader program is usually run from a

TS01-1542

Central Monitor (CM) program that provides a Graphical User Interface (GUI) which gives a "bird's eye view" of the manufacturing Fab real-time status. There are usually two major steps within a loader program. Step one involves calculating the most updated data at the source RDBMS tables. In step two, the loader program processes a pre-summary portion of the data and updates the target RDBMS tables with the newest data. One issue with the loader program is its execution performance. This is because in most cases the loader program must spend system resources and time to maintain the data at the target RDBMS. Therefore, if the results of the first step are not optimized, the second step's efficiency will be impacted resulting in poor overall performance for the loader program and slow data synchronization between source and target RDBMS's which might not be real time enough for the user's needs.

The current design of the CM loader program has one serious drawback. When the total number of records at the target the RDBMS becomes increasingly larger, such as when Work In Process (WIP) volume increases, the performance of step one will heavily slow down the overall performance of the CM loader program. This invention's method and system solve this performance problem.

In US 6,345,259 (Sandoval) a system and method for the integrating of business manufacturing environments is discussed. In 5,880,960 (Lin et al.) a method to improve WIP balance in a manufacturing line is discussed. In US 5,612,886 (Weng) a method and system for dynamic dispatching is discussed. In US 5,463,555 (Ward et al.) a system and method for integrating a

business environment with a process control environment is discussed.

### SUMMARY OF THE INVENTION

This invention's overall objective is to provide a method of improving the performance of relational databases when doing data replication from a source database to a target database. It is a more specific objective to eliminate the need for the target database to be involved in calculating the new target's data. Another objective is to quickly perform an analysis by using the time and date stamp of the source database records to determine which data has changed. Yet other objectives are to perform the data reduction by deleting the changed records from the target data, and then inserting the updated changed records into the target database without any other further time-consuming checks necessary. Thus, the overall computer processing necessary to accomplish the data reduction is greatly reduced and performance is increased.

### BRIEF DESCRIPTION OF THE DRAWINGS

This invention will be described with reference to the accompanying drawings, wherein:

Fig. 1 is a flow diagram of the prior art method of keeping a real-time production database

TS01-1542

current.

Fig. 2 is a flow diagram of the new method of database updating.

Fig. 3 is an example block diagram showing initial state of a database table structure.

Fig. 4A, 4B, and 4C are example block diagrams showing the steps in changing the table structure of the new method.

### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

In today's manufacturing systems where numerous tools are needed to build a product, control of the system is essential. To achieve this, computer systems are used to monitor and run many of the tools as the speed, processing, and complexity are too much for individual operators to manage on a continual basis. This is especially true in the manufacture of semiconductor wafers and chips where product is processed in a strict sequence and any error can produce very expensive scrap. The ability to monitor this production in close to real time is essential for efficiency, quality, and cost effectiveness.

To monitor production in real time using a computer system, data must be gathered in real time by the system tools and processes. The most updated data at source RDBMS tables is calculated, and then a pre-summary portion is processed and displayed on a control monitor and the newest data is updated into target RDBMS tables. Execution performance can be an issue for the loader

TS01-1542

program as system resources and time must be spent to maintain data at the target RDBMS.

Overall performance can be negatively impacted when the calculation of the most updated data at source RDBMS tables is not optimized. For example, with the current design of a CM loader program when the total count of records at the target RDBMS becomes larger (such as when WIP volume increases), the calculating of the most updated data requires more comparison time due to the increase in the number of records. This will slow down the overall performance of the CM loader program.

Fig. 1 shows the prior art method for keeping data current in a RDBMS. In the first block 102, all target data `eqp_id + lot_stat+lot_id` (a combination of equipment ID, lot status, and lot ID parameters) is gotten and stored in memory. Then the data is compared 104 with original source data. A check is done to find out which `eqp_id + lot_stat+lot_id` combinations existed in the source data 106. If no (such as a lot that was changed from one tool to another), this now out-of-date data is deleted from target data 108. If yes, the updated changed records are gotten from the source data 110 and inserted into the target data 112.

This process, although seemingly simple, in a real-time environment requires too much computer resource processing time and memory. As either the source or target table grows, both the procuring step 102 and comparing step 104 will take longer, perhaps so much longer that the data displayed is no longer in the immediate time frame that is necessary. Clearly, another method that has more efficient logic to maintain and synchronize data is necessary for better results. This

invention optimizes the process and solves the problems.

The new method of replication is simpler and requires much less computer resources, thus enabling more effective real-time operation. A loader program provides the computing power for the replication and is capable of displaying the manufacturing equipment environment and lot status on a central monitor. In Fig. 2, a block diagram of the new method flow is shown applied to an equipment WIP list where many lots can be associated with one piece of equipment. The equipment WIP list is started at the source RDBMS. The new method analyzes time and date stamp to determine if a record in the source database has been changed 22. A changed record results when the position of a lot is changed from one piece of equipment to another and includes the record that is no longer valid and its update. When the time and date stamp indicate a change, it locates the record according to its lot identifier in the source database and deletes the changed outdated record from the target data 24 thereby reducing the amount of data. The updated changed record is then inserted into the target data 26. This is called a Delete Changed First approach and eliminates the need for the target database to be involved in calculating new target data as only changed records in the source database are looked at. This conserves computer resources by being source data volume independent and eliminating the need to compare tables. This conservation of computer resources and the execution performance being independent of the volume of source data allows for real-time synchronization between the source equipment WIP list and the target equipment WIP list. Replicated data has the capability of being exported to another database or software system.

TS01-1542

Now let us look at an example of how the records are actually changed. A source table that can be thought of as the master table contains the current WIP list of lots waiting for each piece of equipment or tool. It is maintained in real time by the Manufacturing Execution System (MES) which controls the entire manufacturing floor operation. To monitor the actual WIP lots, we need to periodically read the source table that is contained in an MES-controlled RDBMS. Since all the data contained in that database table is not needed for our display system, we create a simpler target data table to use for the GUI display program or central monitor.

Assume the current situation in source and target data tables is as in Fig. 3 for two tools called Tool A and Tool B. It shows the WIP lots that are waiting (WAIT) to be processed at each tool. The source tables' content 32 34 is identical to the target tables' content 36 38 at our starting instance. Tool A has Lot D waiting to be processed 32, and Tool B has lots A, B, and C waiting to be processed 34.

If an MES user performs a move operation on Tool B to move Lot B to another tool and adds a Lot D to Tool B 42 as shown in Fig. 4A, we now have two records in the source database table that are changed records. This is noted automatically in their records date and time stamp and leads us to an important observation. Since all lot movement will create a noted change in the date and time stamp of the record, it can simplify the method of updating the target database. If we look to Delete Changed First as our first step, the no-longer-valid records can be removed with no further checking or processing required, thus eliminating the time and resource

TS01-1542

consuming steps of the prior art method.

After deleting the changed records quickly from the target data table, the target data table looks like that of Fig. 4B. In this figure, Tool A 44 has had the Lot D removed and Tool B 46 has had Lot B removed. The updated changed records are now inserted into the target table. Lot D was moved from Tool A 48 to Tool B 49. Thus, Fig. 4C shows the now updated target table with Lot D now waiting in Tool B. The target table that Lot B was moved to would subsequently be updated. These updates are simple, fast, and allow the synchronization to take place in real time, thereby maintaining good performance. No other processing is required. The target becomes the source for the subsequent processing and the method is repeated again and again allowing for fast real-time display of the data with minimal computer resources used.

The method of the invention provides advantages over the prior art including making it possible to meet the real-time synchronization requirement at the loader program, speeding up generating performance and maintaining it even when source and target data grows, and using one source data changed record set to determine no longer valid records for deletion and insertion of updated records. The simpler, optimized steps of this invention eliminates the need for a comparison loop, thus conserving computer time and resources.

While the invention has been particularly shown and described with reference to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form



**TS01-1542**

**and details may be made without departing from the spirit and scope of the invention.**

**What is claimed is:**